

14<sup>th</sup> Panhellenic Logic Symposium, Thessaloniki, 1-5 July 2024

## TUTORIAL

Part I: Formalising mathematics with proof assistants

Part II: Getting started with Isabelle/HOL & bonus example: Aristotle's  
Assertoric Syllogistic in Isabelle/HOL

Angeliki Koutsoukou-Argyraiki

Royal Holloway, University of London, UK  
and  
University of Cambridge, UK



# My memories of the PLS

- \* Athens 2013 (National Technical University of Athens)
- \* Samos 2015

The University of Aegean  
the University of Crete  
and  
the University of Athens  
co-organize the

10ο Πανελλήνιο Συμπόσιο Λογικής  
(10th Panhellenic Logic Symposium)

Invited Speakers

- J.-Y. Beziau (University of Rio de Janeiro, Brazil)
- L. Crosilla (University of Leeds, UK)
- P. D'Aquino (University of Napoli II)
- V. Gregoriades (TU Darmstadt, Germany)
- R. Sklinos (University of Lyon 1, France)
- M. Soškova (Sofia University, Bulgaria)
- N. Tzevelekos (Queen Mary University of London, UK)
- X. Vidaux (University of Concepcion, Chile)

Organizing Committee

- Charalampos Cornarós (Chair)
- Costas Dimitracopoulos
- Nikolaos Pappaspyrou

Web site  
[samosweb.aegean.gr/pls10](http://samosweb.aegean.gr/pls10)

SPONSORS

June 11 - June 15 2015  
Samos, Greece

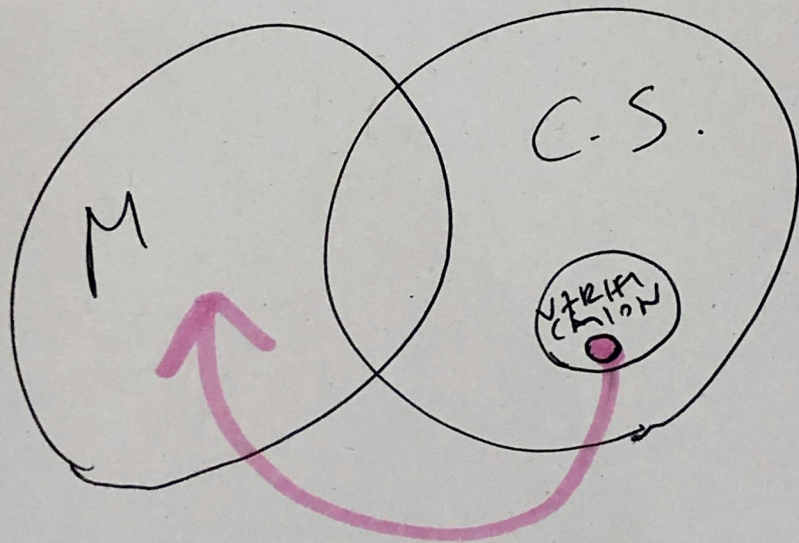
ΑΤΚ & ΤΡΑΝΣΠΟΡΤΣ Α.Ε.  
ΕΛΛΗΝΙΚΗ ΑΕΡΟΠΟΡΙΑ  
Blue Star Ferries  
ΕΥΡΩΠΑΪΚΗ  
TAKIS  
Shop & Go  
ΕΛΛΗΝΙΚΕΣ ΣΕΙΣ  
OZUCOR  
Αεροπλάνο  
cinema  
ΠΑΡΑΡΤΗΜΑΤΑ

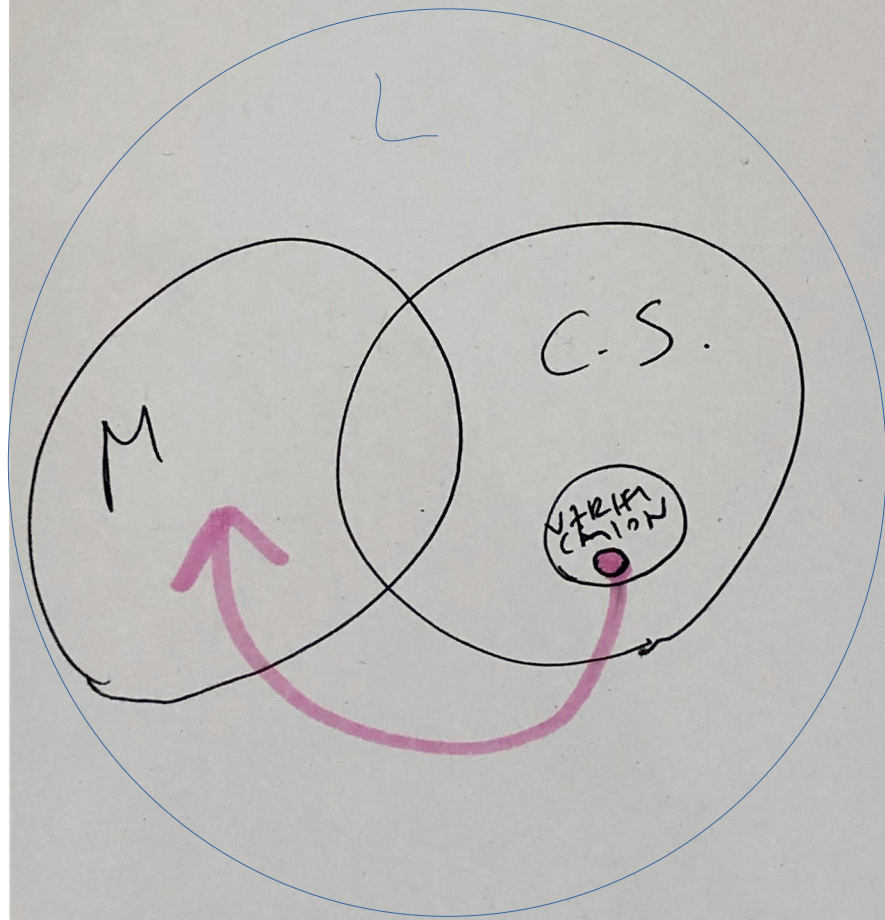


# My memories of the PLS

\* Delphi 2017







# A bit of history

## Leibniz (1666)

“Dissertatio de arte combinatoria”: proposes the development of a symbolic language that could express any rational thought (*characteristica universalis*) and a mechanical method to determine its truth (*calculus ratiocinator*). To resolve any dispute: “Let us calculate!”/ “*Calculemus!*”

## Boole (1847)

“The mathematical analysis of logic”: propositional logic.

## Frege (1879)

“*Begriffsschrift*”: an expressive formal language equipped with logical axioms and rules of inference.

# A bit of history

## Whitehead and Russell (1910-1913)

“Principia Mathematica”: (logicism) goal to express all mathematical propositions in symbolic logic & solve paradoxes of set theory. Developed type theory.

## Hilbert (1920)

Formalism and Hilbert’s program: All mathematical statements should be written in a precise formal language, follow from a provably consistent finite system of axioms, according to well-defined rules. Completeness, Consistency, Conservation, Decidability.

## Note: Gödel’s Incompleteness Theorems (1931)

# A bit of history

## de Bruijn (late 1960s)

AUTOMATH: a predecessor of modern proof assistants based on type theory. Used Curry–Howard correspondence. Late 1970's: van Benthem Jutting translated Landau's "Foundations of Analysis" into AUTOMATH.

## The QED Manifesto (1994)

A proposal for a central computer-based library of all known mathematics fully formalised and formally verified (automatically checked by computers)

The project was soon abandoned.

**(Or was it?)**



# The QED Manifesto (1994)

A proposal for a central computer-based library of all known mathematics fully formalised and formally verified (automatically checked by computers).

The QED Manifesto\*

May 15, 1994

*The development of mathematics toward greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules.*

– K. Gödel

*If civilization continues to advance, in the next two thousand years the overwhelming novelty in human thought will be the dominance of mathematical understanding.*

– A. N. Whitehead

of all, or even of the most important, mathematical results something beyond the capacity of any human. For example, few mathematicians, if any, will ever understand the entirety of the recently settled structure of simple finite groups or the proof of the four color theorem. Remarkably, however, the creation of mathematical logic and the advance of computing technology have also provided the means for building a computing system that represents all important mathematical knowledge in an entirely rigorous and mechanically usable fashion. The QED system we imagine will provide a means by which mathematicians and scientists can scan the entirety of mathematical knowledge for relevant results and, using tools of the QED system, build upon such results with reliability and confidence but without the need for minute comprehension of the details or even the ultimate foundations of the parts of the system upon which they build.

The project was soon abandoned.

**(Or was it?)**

## 1 What Is the QED Project and Why Is It Important?

QED is the very tentative title of a project to

# Today

## Modern proof assistants (interactive theorem provers)

Software tools for formal verification/ the development of formal proofs by user-computer interaction. A human user writes the proof in a formal language via an interactive interface to be checked by a computer. Intermediate proof steps are often given by automation.

A variety of proof assistants available, based on different logical formalisms:  
Based on: set theory (e.g. Mizar, Metamath); simple type theory (e.g. HOL4, HOL Light, Isabelle); dependent type theory (e.g. Coq, Agda, Lean, PVS).  
Extensive libraries of formalised mathematics available.

For a direct comparison with examples, see, e.g. the webpage maintained by Wiedijk, “Formalising 100 theorems”.

# Why formalise mathematics?

**...a comment on my original personal motivation: insights into the nature of proofs**

Work in applied proof theory/proof mining: pen-and-paper extraction of constructive/quantitative information from proofs in the form of computable bounds...

...Provokes the question:

What is it that makes a “good” proof?

- \* a shorter proof;
- \* a more “elegant” proof;
- \* a simpler proof (consider Hilbert’s 24th problem (1900)): “find criteria for simplicity of proofs, or, to show that certain proofs are simpler than any others.”;
- \* in terms of Reverse Mathematics – a proof in a weaker subsystem of Second Order Arithmetic;
- \* an interdisciplinary proof (e.g. a geometric proof for an algebraic problem or vice-versa would be considered to give a deeper mathematical insight);
- \* a proof that is easier to reuse i.e. if it provides some algorithm or technique or intermediate result that can be useful in different contexts too;

- \* a proof giving “better” computational content.

What do we mean by “better” computational content?

- \* a bound of lower complexity?

- \* a bound that is more precise numerically?

- \* a bound that is more “elegant”?

# Why formalise mathematics?

\* Verification: eliminating mathematical mistakes (Example: the Fields medalist Vladimir Voevodsky started working in formalisation after discovering errors in his own work).

## The Origins and Motivations of Univalent Foundations

(2014)

*Professor Voevodsky's Personal Mission to Develop Computer Proof Verification to Avoid Mathematical Mistakes*

BY VLADIMIR VOEVODSKY

In January 1984, Alexander Grothendieck submitted to the French National Centre for Scientific Research his proposal “Esquisse d’un Programme.” Soon copies of this text started circulating among mathematicians. A few months later, as a first-year undergraduate at Moscow University, I was given a copy of it by George Shabat,

is hardly ever checked in detail.

But this is not the only problem that allows mistakes in mathematical texts to persist. In October 1998, Carlos Simpson submitted to the arXiv preprint server a paper called “Homotopy Types of Strict 3-groupoids.” It claimed to provide an argument that implied that the main result of the “ $\infty$ -groupoids” paper, which Kapranov and I had published in 1989, cannot be true. However, Kapranov and I had consid-

# How to write a 21<sup>st</sup> century proof

Leslie Lamport

*To D. Palais*

**Abstract.** A method of writing proofs is described that makes it harder to prove things that are not true. The method, based on hierarchical structuring, is simple and practical. The author's twenty years of experience writing such proofs is discussed.

**Mathematics Subject Classification (2010).** 03B35, 03F07.

**Keywords.** Structured proofs, teaching proofs.

In addition to developing the students' intuition about the beautiful concepts of analysis, it is surely equally important to persuade them that precision and rigor are neither deterrents to intuition, nor ends in themselves, but the natural medium in which to formulate and think about mathematical questions.

Michael Spivak, *Calculus* [7]

# Why formalise mathematics?

\* (Future of?) Reviewing.

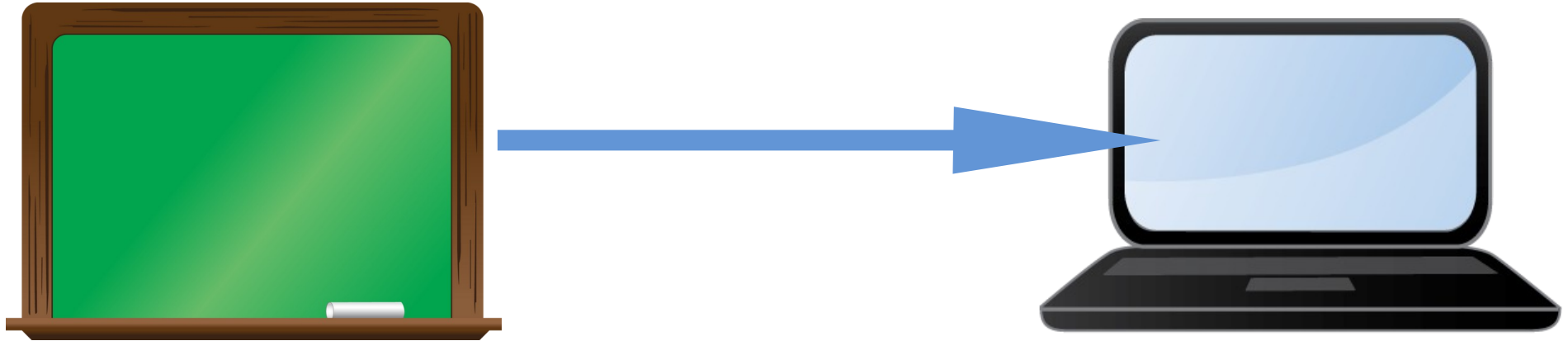
\* Preserving mathematical knowledge in big libraries of formalised mathematics: databases with an enormous potential for the creation of future AI tools to assist mathematicians in the discovery(/invention) of new results.



# Why formalise mathematics?

\* Deeper understanding, new insights: even familiar material can be seen in a new light when using new tools. High level of detail in which a formalised proof must be written forces to think and rethink proofs and definitions.

## The computer as a “magic mirror”



# Why formalise mathematics?

- \* A way of keeping track of all the details of a complicated proof.

the other way around! The Lean Proof Assistant was really that: An assistant in navigating through the thick jungle that this proof is. Really, one key problem I had when I was trying to find this proof was that I was essentially unable to keep all the objects in my “RAM”, and I think the same problem occurs when trying to read the proof. Lean always gives you a clear formulation of the current goal, and Johan confirmed to me that when he formalized the proof of Theorem 9.4, he could — with the help of Lean — really only see one or two steps ahead, formalize those, and then proceed to the next step. So I think here we have witnessed an experiment where the proof assistant has actually assisted in understanding the proof.

Peter Scholze (Fields Medal 2018)  
June 2021, Xena Project Blog

The Blueprint tool by Patrick Massot for Lean shows interdependency of proof parts, tracks formalisation progress.

- \* Educational tools.
- \* Last but not least: it is fulfilling and fun!

# A vision for the future of research mathematics:

To create an interactive assistant that would help research mathematicians in their creative work by

- \* providing “brainstorming”/ hints:  
proof recommendations, counterexamples, proofs of auxiliary lemmas/intermediate steps;
- \* suggesting conjectures;
- \* providing information on relevant literature results;
- \* helping with bookkeeping on the proof structure/proof goals and details;
- \* formally verifying the new results.

The goal is to assist mathematicians, not to replace them.

# A vision for the future of research mathematics:

Timothy Gowers (Fields Medal 1998) describes how a "dialogue" between a user and a computer would ideally look like in the future to interactively assist the human mathematician to arrive at (new) conclusions. The computer would have access to an extensive database of mathematical material.

W.T. Gowers (2010). Rough Structure and Classification. In: Alon, N., Bourgain, J., Connes, A., Gromov, M., Milman, V. (eds) Visions in Mathematics. Modern Birkhäuser Classics. Birkhäuser Basel. [https://doi.org/10.1007/978-3-0346-0422-2\\_4](https://doi.org/10.1007/978-3-0346-0422-2_4)

*“We believe that when later generations look back at the development of mathematics one will recognise four important steps:*

*(1) the Egyptian-Babylonian-Chinese phase, in which correct computations were made, without proofs;*

*(2) the ancient Greeks with the development of “proof”;*

*(3) the end of the nineteenth century when mathematics became “rigorous”;*

*(4) the present, when mathematics (supported by computer) finally becomes fully precise and fully transparent.”*



Barendregt, H. and Wiedijk, F. (The challenge of computer mathematics, Philos. Trans. - Royal Soc., Math. Phys. Eng. Sci. 36(1835):2351-2375 (2005)).

# Towards a new era in Mathematics?

A big shift: Formalisation was until recently an area of computer science. Now it is quickly attracting the interest of working mathematicians and mathematics students too. Enthusiastic online communities and tools e.g. Zulip enable massive collaborative projects. Libraries of formal proofs are expanding at an increasingly high pace, day-by-day. Student-run projects are emerging too. Everyone welcome to join.

\* The 2020 Mathematics Subject Classification includes for the first time subject classes on the formalisation of mathematics using proof assistants (68VXX).

\* Kevin Buzzard and Georges Gonthier invited at the 2022 International Congress of Mathematicians to talk about the formalisation of mathematics.

\* May-August 2024: Trimester Program “Prospects of Formal Mathematics”, Hausdorff Research Institute for Mathematics, Bonn.

# Some milestones & recent advances

- \* Formalisation of the proof of the four-colour theorem in Coq by Gonthier (2008).
- \* Gonthier has also formalised the Feit–Thompson proof of the odd-order theorem in Coq (2012).
- \* Formalisation of the proof (1998 publ. 2005) by Hales of the Kepler conjecture (sphere packing problem) in HOL Light and Isabelle/HOL by Hales et al. (Flyspeck project, 2003-compl. 2014).
- \* Formalisation of Gödel's Incompleteness theorems in Isabelle/HOL by Paulson (2013).

# Some milestones & recent advances

- \* Formalisation of an irrationality proof of  $\zeta(3)$  by Apéry (evaluation of the Riemann zeta function) in Coq by Chyzak, Mahboubi, Sibut-Pinote & Tassi (2014).
- \* Verification of an algorithm with Isabelle/HOL to verify Tucker's proof that the Lorenz attractor is chaotic in a rigorous mathematical sense by Immler (2015).
- \* Formalisation of Scholze's perfectoid spaces in Lean by Buzzard, Commelin and Massot (2019).
- \* Grothendieck's schemes in Lean by Buzzard, Hughes, Lau, Livingston, Fernández Mir, R., Morrison, S. (2020).  
Independently in Isabelle/HOL by Bordg, Li and Paulson (2021).



# Some milestones & recent advances

- \* Formalisation of a substantial amount of material in analytic number theory in Isabelle/HOL by Eberl (2019), Eberl, Paulson, Bordg and Li (2023).
- \* The independence of the Continuum Hypothesis by Han & van Doorn in Lean (2021). Independently in Isabelle/ZF by Gunther, Pagano, Sánchez Terraf & Steinberg (2022).
- \* Formalisation of the solution to the cap set problem (Ellenberg & Gijswijt, 2017) by Dahmen, Hölzl and Lewis in Lean (2019).
- \* Szemerédi's Regularity Lemma and Roth's Theorem on Arithmetic Progressions in Isabelle/HOL by Edmonds, Koutsoukou-Argyraki and Paulson. Independently in Lean by Dillies and Mehta (2021).

\* Formalising Szemerédi's Regularity Lemma and Roth's Theorem on Arithmetic Progressions in Isabelle/HOL (Chelsea Edmonds, A. K.-A. & Lawrence C. Paulson, Journal of Automated Reasoning, vol. 67, Article number: 2 (2023), online 19/12/2022.)

Fundamental results in extremal graph theory and combinatorics/number theory. (simultaneously and independently formalised in Lean by Mehta and Dillies)

AFP entries:

- Roth's Theorem on Arithmetic Progressions (Edmonds, A. K.-A. & Paulson, 2021).
- Szemerédi's Regularity Lemma (Edmonds, A. K.-A. & Paulson, 2021).

Main sources: book by Y. Zhao, notes from course by W. T. Gowers.

The *upper asymptotic density* of a set  $A \subseteq \mathbb{Z}$  is defined as

$$\limsup_{N \rightarrow \infty} \frac{|A \cap [1, N]|}{N}.$$

## Szemerédi (1975)

Every set of integers  $A$  with positive upper asymptotic density contains a  $k$ -term arithmetic progression for every  $k \in \mathbb{N}$ .

## Roth (1953)

Every subset of the integers with positive upper asymptotic density contains a 3-term arithmetic progression.

**theorem** RothArithmeticProgressions:

**assumes** "upper\_asymptotic\_density  $A > 0$ "

**shows** " $\exists k d. d > 0 \wedge \text{progression3 } k d \subseteq A$ "

For sets of vertices  $X, Y \subseteq V(G)$ , let  $e(X, Y)$  be the number of edges between  $X$  and  $Y$ . That is,

$$e(X, Y) = |\{(x, y) \in X \times Y : xy \in E(G)\}|.$$

Given a graph  $G$ , for sets of vertices  $X, Y \subseteq V(G)$ , we define the edge density between  $X$  and  $Y$  to be

$$d(X, Y) = \frac{e(X, Y)}{|X||Y|}.$$

Given a graph  $G$  and  $\epsilon > 0$ , for sets of vertices  $X, Y \subseteq V(G)$ , we call  $(X, Y)$  an  $\epsilon$ -regular pair (in  $G$ ) if for all  $A \subseteq X$ ,  $B \subseteq Y$  with  $|A| \geq \epsilon|X|$ ,  $|B| \geq \epsilon|Y|$ , one has

$$|d(A, B) - d(X, Y)| \leq \epsilon.$$

Given a graph  $G$  and  $\epsilon > 0$ , a partition  $P = \{V_1, \dots, V_k\}$  of  $V(G)$  is an  $\epsilon$ -regular partition if

$$\sum_{\substack{(i,j) \in [k]^2 \\ (V_i, V_j) \text{ not } \epsilon\text{-regular}}} |V_i||V_j| \leq \epsilon|V(G)|^2.$$

## Szemerédi (1975) Regularity Lemma

For every  $\epsilon > 0$ , there exists a constant  $M$  such that every graph has an  $\epsilon$ -regular partition of its vertex set into at most  $M$  parts.

**theorem** Szemerédi\_Regularity\_Lemma:

**assumes** " $\epsilon > 0$ "

**obtains**  $M$  **where** " $\wedge G. \text{card}(\text{verts } G) > 0 \implies \exists P. \text{regular\_partition } \epsilon G P \wedge \text{card } P \leq M$ "

## Triangle Counting Lemma

Given a graph  $G$ , let  $X, Y, Z \subseteq V(G)$  so that  $(X, Y), (Y, Z), (Z, X)$  are all  $\epsilon$ -regular pairs for some  $\epsilon > 0$ . Assuming that  $d(X, Y), d(X, Z), d(Z, Y) \geq 2\epsilon$ , the number of triples  $(x, y, z) \in X \times Y \times Z$  such that  $x, y, z$  form a triangle in  $G$  is at least

$$(1 - 2\epsilon)(d(X, Y) - \epsilon)(d(X, Z) - \epsilon)(d(Y, Z) - \epsilon)|X||Y||Z|.$$

**theorem** triangle\_counting\_lemma:

**fixes**  $\epsilon :: \text{real}$

**assumes** xss: " $X \subseteq \text{uverts } G$ " **and** yss: " $Y \subseteq \text{uverts } G$ " **and** zss: " $Z \subseteq \text{uverts } G$ " **and** en0: " $\epsilon > 0$ "

**and** finG: "finite (uverts  $G$ )" **and** wf: "uwellformed  $G$ "

**and** rp1: "regular\_pair  $X Y G \epsilon$ " **and** rp2: "regular\_pair  $Y Z G \epsilon$ " **and** rp3: "regular\_pair  $X Z G \epsilon$ "

**and** ed1: "edge\_density  $X Y G \geq 2*\epsilon$ " **and** ed2: "edge\_density  $X Z G \geq 2*\epsilon$ " **and** ed3: "edge\_density  $Y Z G \geq 2*\epsilon$ "

**shows** "card (triangle\_triples  $X Y Z G$ )

$\geq (1 - 2*\epsilon) * (\text{edge\_density } X Y G - \epsilon) * (\text{edge\_density } X Z G - \epsilon) * (\text{edge\_density } Y Z G - \epsilon) *$

card  $X$  \* card  $Y$  \* card  $Z$ "

# Triangle Removal Lemma

For all  $\epsilon > 0$ , there exists  $\delta > 0$  such that any graph on  $N$  vertices with less than or equal to  $\delta N^3$  triangles can be made triangle-free by removing at most  $\epsilon N^2$  edges.

**theorem** `triangle_removal_lemma`:

**fixes**  $\epsilon :: \text{real}$

**assumes** `egt`: " $\epsilon > 0$ "

**shows** " $\exists \delta :: \text{real} > 0. \forall G. \text{card}(\text{uverts } G) > 0 \longrightarrow \text{uwellformed } G \longrightarrow$

$\text{card}(\text{triangle\_set } G) \leq \delta * \text{card}(\text{uverts } G) ^ 3 \longrightarrow$

$(\exists G'. \text{triangle\_free\_graph } G' \wedge \text{uverts } G' = \text{uverts } G \wedge \text{uedges } G' \subseteq \text{uedges } G \wedge$

$\text{card}(\text{uedges } G - \text{uedges } G') \leq \epsilon * (\text{card}(\text{uverts } G) ^ 2)$ "

**(is** " $\exists \delta :: \text{real} > 0. \forall G. \_ \longrightarrow \_ \longrightarrow \_ \longrightarrow (\exists G_{\text{new}}. ?\Phi G G_{\text{new}})$ "

# Some milestones & recent advances

- \* Massot, van Doorn and Nash formalised in Lean results in differential topology on sphere eversion (2021).
- \* Mehta recently formalised in Lean a 2023 result by Campos, Griffiths, Morris and Sahasrabudhe on an exponential improvement to the upper bound on Ramsey numbers.
- \* Mehta and Bloom formalised (2022) in Lean a 2021 paper by Bloom on unit fractions.
- \* A formalisation of the Balog–Szemerédi–Gowers Theorem in Isabelle/HOL by Koutsoukou-Argyraiki, Bakšys & Edmonds (2022).



## Basic definitions:

Let  $A, B$  be finite subsets of an abelian group. The *sumset*  $A + B$  is the set  $\{a + b \mid a \in A, b \in B\}$ . The *difference set*  $A - B$  is the set  $\{a - b \mid a \in A, b \in B\}$ . For  $n$  many copies  $A + \dots + A$  we write  $nA$ .

Let  $G$  be an abelian group. An *additive quadruple* in  $G$  is a quadruple  $(a, b, c, d) \in G^4$  such that  $a + b = c + d$ . The *additive energy* of a subset  $A$  of  $G$  is the number of additive quadruples in  $A^4$  divided by  $|A|^3$ .

\* A formalisation of the Balog–Szemerédi–Gowers Theorem in Isabelle/HOL (A. K.-A., Mantas Bakšys & Chelsea Edmonds, in CPP '23: 12th ACM SIGPLAN, International Conference on Certified Programs and Proofs ).

A profound result in additive combinatorics which played a central role in Gowers's proof deriving the first effective bounds for Szemerédi's Theorem on arithmetic progressions.

**Balog & Szemerédi (1994):** Every finite subset of an abelian group of given additive energy must contain a large subset whose sumset is small.

**Gowers (2001):** New proof with better bounds on the cardinalities.

## Balog-Szemerédi-Gowers:

Let  $A$  be a finite subset of an abelian group. Suppose that  $A$  has additive energy  $2c$  for some  $c > 0$ . Then  $A$  has a subset  $A'$  so that  $|A'| \geq c^2|A|/4$  and  $|A' - A'| \leq 2^{30}|A|/c^{34}$ .

```
theorem Balog_Szemerédi_Gowers: fixes A::"a set" and c::real
  assumes afin: "finite A" and "A ≠ {}" and "c>0" and "additive_energy A = 2 * c" and ass: "A ⊆ G"
  obtains A' where "A' ⊆ A" and "card A' ≥ c^2 * card A / 4" and
    "card (differenceset A' A') ≤ 2^30 * card A / c^34"
```

(Analogous version for sumsets).

The proof involves a fascinating interplay between graph theory, probability theory, additive combinatorics: expressed via an implementation of locales, Isabelle's module system.

Made use of a new, general undirected graph theory library by Edmonds.

# Some milestones & recent advances

## Student activity

\* **Kevin Buzzard's London Lean community at Imperial College London (Xena Project)**

\* A group of undergraduate students formalised in Isabelle/HOL  
Matiyasevich's proof of the DPRM theorem (1970):  
every recursively enumerable set of natural numbers is Diophantine. This  
gives a negative solution to Hilbert's 10th problem over the integers.

AFP entry:

-Diophantine Equations and the DPRM Theorem

(Jonas Bayer, Marco David, Benedikt Stock, Abhik Pal, Yuri Matiyasevich  
and Dierk Schleicher, 2022)

# Some milestones & recent advances

## The Liquid Tensor Experiment

Condensed Mathematics (Clausen and Scholze) introduces condensed sets, an alternative notion to topological spaces. Scholze posed a formalisation challenge (Xena Project Blog, Dec. 2020) in this area. The Lean Prover Community took up the challenge: a huge collaborative effort led by Commelin succeeded in formalising the proof in July 2022.



PROOFS

### Proof Assistant Makes Jump to Big-League Math

7 |

Mathematicians using the computer program Lean have verified the accuracy of a difficult theorem at the cutting edge of research mathematics.



[nature](#) > [news](#) > article

NEWS | 18 June 2021

# Mathematicians welcome computer-assisted proof in ‘grand unification’ theory

**Proof-assistant software handles an abstract concept at the cutting edge of research, revealing a bigger role for software in mathematics.**

[Davide Castelvechi](#)



# Some milestones & recent advances

## Important developments coming up

\* Buzzard is starting in 2024 a new 5-year project to formalise in Lean much of the mathematics involved in the proof of Fermat's Last Theorem.

\* Terence Tao (Fields Medal 2006) announced on his blog (13/11/23) that he is planning to formalize in Lean, together with Dillies and Mehta, the new Gowers–Green–Manners–Tao proof of the Polynomial Freiman–Ruzsa conjecture (first proposed by Katalin Marton).

The project was completed in about 3 weeks thanks to a group of Lean contributors.

# The ALEXANDRIA Project at Cambridge (2017-2023)

“Large Scale Formal Proof for the Working Mathematician”  
led by Professor Lawrence C. Paulson FRS

<https://www.cl.cam.ac.uk/~lp15/Grants/Alexandria/>

- Expanding the body of formalised material on the Archive of Formal Proofs and the Isabelle Libraries.
- Case studies to explore the limits of formalisation.
- Tools for managing large bodies of formal mathematical knowledge (intelligent search/ computer-aided knowledge discovery).
- Automated and semi-automated environments and tools to aid *working mathematicians*.

Postdocs: Wenda Li, Anthony Bordg, Yiannos Stathopoulos,  
Angeliki Koutsoukou-Argyraiki. PhD Student: Chelsea Edmonds.  
Many external collaborators and interns.



UNIVERSITY OF  
CAMBRIDGE



European Research Council  
Established by the European Commission







[International Conference on Intelligent Computer Mathematics](#)

↳ CICM 2023: [Intelligent Computer Mathematics](#) pp 3–15 | [Cite as](#)

[Home](#) > [Intelligent Computer Mathematics](#) > [Conference paper](#)

## Large-Scale Formal Proof for the Working Mathematician—Lessons Learnt from the ALEXANDRIA Project

[Lawrence C. Paulson](#) 

Conference paper | [First Online: 28 August 2023](#)

79 Accesses

Part of the [Lecture Notes in Computer Science](#) book series (LNAI, volume 14101)



UNIVERSITY OF  
CAMBRIDGE



European Research Council

Established by the European Commission



# References on selected contributions of mine within ALEXANDRIA

Summarized in recent talks:

\* In mid-2022 I initiated a line of work to formalise material in additive combinatorics, on the structure of sumsets of finite subsets of abelian groups. (See my invited talk in the proceedings of the 14<sup>th</sup> Conference on Interactive Theorem Proving (ITP 2023)

DOI: 10.4230/LIPIcs.ITP.2023.1)

\* See the slides for my two tutorials in Interactions of Proof Assistants and Mathematics, International Summer School, Regensburg, Germany, Sept. 18-29, 2023.



UNIVERSITY OF  
CAMBRIDGE



European Research Council

Established by the European Commission



# Conclusion: Lessons learned so far

- \* Formalisation goals accomplished
- \* Still yet to encounter any material impossible to formalise in simple type theory
- \* Advanced mathematics within reach
- \* Locales can be very useful (to capture interaction between different mathematical areas and to “cheat” by including unformalised material as assumptions)
- \* The formalisation process can reveal the need for a higher level of abstraction in prerequisites.

# Conclusion: Lessons learned so far

- \* Sledgehammer's automation (Isabelle) is practical and efficient
- \* Students can learn Isabelle very fast and formalise advanced material successfully
- \* Collaborative work, filling in library gaps
- \* We still need: better automation, efficient organisation and management of libraries (definitions, elementary properties and basics, advanced results)
- \* Our libraries can grow increasingly fast!

# Main Obstacles

- \* Better automation is needed to provide proofs for intermediate proof steps (proofs are analysed in an extremely high level of detail).
- \* Efficient search features.
- \* Efficient organisation and management of libraries.
- \* Readability of formal proofs by humans.
- \* Interoperability of proof systems, translation of proofs between proof assistants needed (Goals of the Dedukti System/ EuroProofNet COST Action).

**Where do we go from here?**

**The future (of mathematics) is hard to  
predict...**

## AI translates maths problems into code to make them easier to solve

An artificial intelligence that can turn mathematical concepts written in English into a formal proving language for computers could make problems easier for other AIs to solve



**MATHEMATICS** 6 June 2022

By [Alex Wilkins](#)

### Autoformalization with Large Language Models

Wu, Y., Jiang, A. Q., Li, W., Rabe, M. N., Staats, C., Jamnik, M., Szegedy, C.  
arXiv:2205.12615v1 in NeurIPS 2022.

---

(OpenAI, 2022)

# Formal Mathematics Statement Curriculum Learning

---

Stanislas Polu<sup>1</sup> Jesse Michael Han<sup>1</sup> Kunhao Zheng<sup>2</sup> Mantas Baksys<sup>3</sup> Igor Babuschkin<sup>1</sup> Ilya Sutskever<sup>1</sup>

## Abstract

We explore the use of expert iteration in the context of language modeling applied to formal mathematics. We show that at same compute budget, expert iteration, by which we mean proof search interleaved with learning, dramatically outperforms proof search only. We also observe that when applied to a collection of formal statements of sufficiently varied difficulty, expert iteration is capable of finding and solving a curriculum of increasingly difficult problems, without the need for associated ground-truth proofs. Finally, by applying this expert iteration to a manually curated set of problem statements, we achieve state-of-the-art on the *miniF2F* benchmark, automatically solving multiple challenging problems drawn from high school olympiads.

whether a trajectory (*i.e.* a proof) is successful (*i.e.* formally correct). But the vast scope of formal mathematics means that any strong reasoning result obtained in it will be more meaningful than comparable results in games (*e.g.* finding proofs to mathematical conjectures), and could even be applicable to important practical problems (*e.g.* software verification).

However, tackling formal mathematics involves two main challenges that we must address in order to continue making progress:

**Infinite action space** Not only does formal mathematics have an extremely large search space (like Go for example), it also has an infinite action space. At each step of proof search, the model must choose not from a well-behaved finite set of actions, but a complex and infinite set of tactics, potentially involving exogenous mathematical terms that have to be generated (*e.g.*, generating a mathematical



\* DeepMind's AI suggests conjectures in research mathematics:  
Machine learning as a mathematical collaborator.

Representation theory:  
Blundell, C., Buesing, L., Davies, A.,  
Veličković, P. Williamson, G.,  
“Towards combinatorial invariance for  
Kazhdan-Lusztig  
polynomials”, arXiv:2111.15161

Not directly related to proof assistants but demonstrates the  
pattern-matching efficiency of AI to assist research mathematicians.

**nature**

[Explore content](#) ▾ [About the journal](#) ▾ [Publish with us](#) ▾

[nature](#) > [articles](#) > article

Article | [Open Access](#) | [Published: 01 December 2021](#)

## Advancing mathematics by guiding human intuition with AI

[Alex Davies](#) ✉, [Petar Veličković](#), [Lars Buesing](#), [Sam Blackwell](#), [Daniel Zheng](#), [Nenad Tomašev](#), [Richard Tanburn](#), [Peter Battaglia](#), [Charles Blundell](#), [András Juhász](#), [Marc Lackenby](#), [Geordie Williamson](#), [Demis Hassabis](#) & [Pushmeet Kohli](#) ✉

[Nature](#) **600**, 70–74 (2021) | [Cite this article](#)

**203k** Accesses | **62** Citations | **1624** Altmetric | [Metrics](#)

[nature](#) > [news](#) > article

NEWS | 01 December 2021

## DeepMind's AI helps untangle the mathematics of knots

The machine-learning techniques could benefit other areas of maths that involve large data sets.

[Davide Castelvecchi](#)



Davies, A., Juhász, A., Lackenby, M., Tomasev, N., The signature and cusp geometry of hyperbolic knots, arXiv:2111.15323v1

(Not related to proof assistants but demonstrates the pattern-matching efficiency of AI to assist research mathematics.)



# Where do we go from here?

## \* Reflection on the concept of mathematical proof and its evolution

### MATHEMATICAL PROOF BETWEEN GENERATIONS

JONAS BAYER <sup>(a)</sup>, CHRISTOPH BENZMÜLLER <sup>(b, a)</sup>, KEVIN BUZZARD <sup>(c)</sup>, MARCO DAVID <sup>(d)</sup>,  
LESLIE LAMPORT <sup>(e)</sup>, YURI MATIYASEVICH <sup>(f)</sup>, LAWRENCE PAULSON <sup>(g)</sup>,  
DIERK SCHLEICHER <sup>(h)</sup>, BENEDIKT STOCK <sup>(i)</sup>, AND EFIM ZELMANOV <sup>(j)</sup>

#### AFFILIATIONS.

- <sup>(a)</sup> Freie Universität Berlin
- <sup>(b)</sup> Otto-Friedrich-Universität Bamberg
- <sup>(c)</sup> Imperial College London
- <sup>(d)</sup> École Normale Supérieure de Paris
- <sup>(e)</sup> Microsoft Research
- <sup>(f)</sup> Steklov Institute of Mathematics at St. Petersburg
- <sup>(g)</sup> University of Cambridge
- <sup>(h)</sup> Aix-Marseille Université
- <sup>(i)</sup> University of Oxford
- <sup>(j)</sup> University of California, San Diego

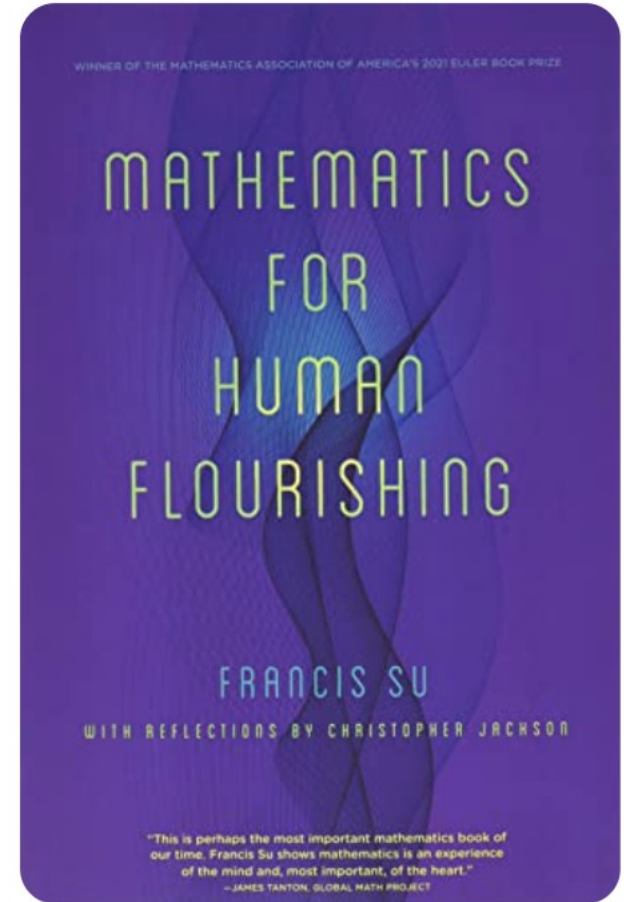
[What Can Formal Systems Do For Mathematics? A Discussion Through The Lens Of Proof Assistants: Some Recent Advances.](#) Q & A with Jeremy Avigad, Jasmin Blanchette, Frédéric Blanqui, Kevin Buzzard, Johan Commelin, Manuel Eberl, Timothy Gowers, Peter Koepke, Assia Mahboubi, Ursula Martin, Lawrence C. Paulson. Invited Contribution. B. Löwe, D. Sarikaya (eds.), 60 Jahre DVMLG, Vol. 48 of Tributes, College Publications, London, 2022.

ABSTRACT. A *proof* is one of the most important concepts of mathematics. However, there is a striking difference between how a proof is defined in theory and how it is used in practice. This puts the unique status of mathematics as exact science into peril. Now may be the time to reconcile theory and practice, i.e. precision and intuition, through the advent of *computer proof assistants*. For the most time this has been a topic for experts in specialized communities. However, mathematical proofs have become increasingly sophisticated, stretching the boundaries of what is humanly comprehensible, so that leading mathematicians have asked for formal verification of their proofs. At the same time, major theorems in mathematics have recently been computer-verified by people from outside of these communities, even by beginning students. This article investigates the gap between the different definitions of a proof and possibilities to build bridges. It is written as a *polemic* or a *collage* by different members of the communities in mathematics and computer science at different stages of their careers, challenging well-known preconceptions and exploring new perspectives.

# Where do we go from here?

“Mathematical exploration is very much like space exploration, but of a different kind of space—a space of ideas. You don’t know what you’ll find when you start. You send out probes to test theories. You are captivated by mystery, motivated by questions, undeterred by setbacks. You make discoveries from a distance: because the ideas themselves are not physical, you access this space through reason. Exploration and understanding are at the heart of what it means to do mathematics [...] Exploration is a deep human desire and a mark of human flourishing.”

Francis Su, *Mathematics for Human Flourishing*, Yale University Press (2020).



# Where do we go from here?

- \* Mathematics is a **profoundly human** activity.
- \* I would **not** claim that every mathematical proof should be formalised to be acceptable
- \* Mathematicians will always seek **understanding**
- \* Hope to use interactive theorem proving to **support** and **assist** mathematical practice

# Where do we go from here?

## **...conceivable to substantially assist with...**

- \* Large collaborative projects
- \* Proof bookkeeping for modern, research-level mathematics
- \* Faster and reliable reviewing for journal submissions
- \* Supporting literature search
- \* Educational support

14<sup>th</sup> Panhellenic Logic Symposium, Thessaloniki, 1-5 July 2024

## TUTORIAL

Part I: Formalising mathematics with proof assistants

Part II: Getting started with Isabelle/HOL & bonus example: Aristotle's  
Assertoric Syllogistic in Isabelle/HOL

Angeliki Koutsoukou-Argyraiki

Royal Holloway, University of London, UK  
and  
University of Cambridge, UK



# Isabelle – A Quick Introduction

Developed by Lawrence C. Paulson (since late 1980's),  
Tobias Nipkow, Makarius Wenzel.

Interactive development of verifiable proofs



(Integrates automated reasoning tools in an interactive setting:

Proof scripts in Isabelle are interactive sessions between user and theorem prover)

- Isabelle/HOL: Higher Order Logic (HOL) (Includes AC; Proofs in classical logic). Simple types.
- Emphasis on producing structured, easy-to-read proofs:

ISAR (Intelligible Semi-Automated Reasoning) proof language.

Internal languages: ML and Scala.

- Features efficient automation (Sledgehammer and counterexample-finding tools like nitpick and Quickcheck).





# Isabelle

[Home](#)[Overview](#)[Installation](#)[Documentation](#)

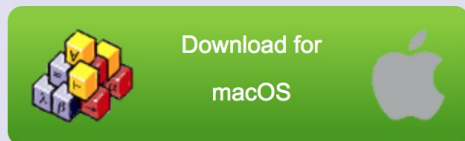
## Site Mirrors:

[Cambridge \(.uk\)](#)  
[Munich \(.de\)](#)  
[Sydney \(.au\)](#)  
[Potsdam, NY \(.us\)](#)

## What is Isabelle?

Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those for logical calculus. Isabelle was originally developed at the [University of Cambridge](#) and [Technische Universität München](#), but now includes numerous contributors from institutions and individuals worldwide. See the [Isabelle overview](#) for a brief introduction.

## Now available: Isabelle2023 (September 2023)



[Download for Linux \(Intel\)](#) - [Download for Linux \(ARM\)](#) - [Download for Windows](#) - [Download for macOS](#)

### Hardware requirements:

- *Small experiments*: 4 GB memory, 2 CPU cores
- *Medium applications*: 8 GB memory, 4 CPU cores
- *Large projects*: 16 GB memory, 8 CPU cores
- *Extra-large projects*: 64 GB memory, 16 CPU cores

### Some notable changes:

- Documents: interactive document preparation via Isabelle/jEdit panel.
- Documents: demos for well-known LaTeX classes.
- Documents: more formal LaTeX citations.
- HOL: various improvements of theory libraries, notably in HOL-Analysis.
- HOL: updates and improvements of Sledgehammer.
- ML: more robust support for ARM64 platform (native Apple Silicon).

# Isabelle – A Quick Introduction

<https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/index.html>

## Isabelle/HOL sessions

### HOL

Classical Higher-order Logic.

### HOL-Algebra

Author: Clemens Ballarin, started 24 September 1999, and many others

The Isabelle Algebraic Library.

### HOL-Analysis

### HOL-Analysis-ex

### HOL-Auth

A new approach to verifying authentication protocols.

### HOL-Bali

### HOL-Cardinals

Ordinals and Cardinals, Full Theories.

### HOL-Codegenerator\_Test

### HOL-Combinatorics

Corecursion Examples.

### HOL-Complex Analysis

### HOL-Computational Algebra

### HOL-Corec Examples

### HOL-Data Structures

Big (co)datatypes.

### HOL-Datatype Benchmark

### HOL-Datatype Examples

(Co)datatype Examples.

### HOL-Decision Procs

Various decision procedures, typically involving reflections



# Isabelle – A Quick Introduction

<https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/HOL-Analysis/index.html>

## Session HOL-Analysis

View [theory dependencies](#)

View [document](#)

View [manual](#)

## Theories

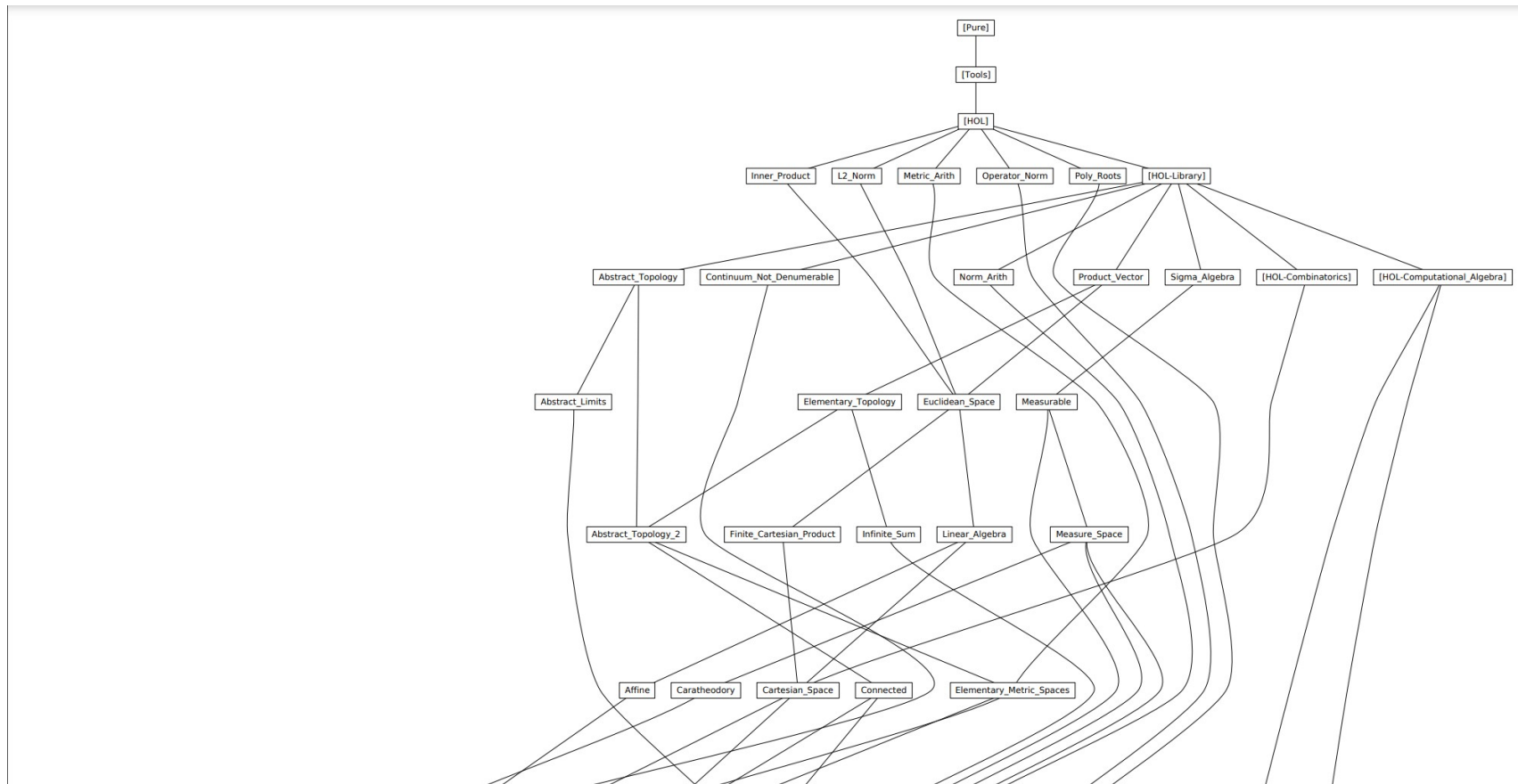
- [L2 Norm](#)
- [Inner Product](#)
- [Product Vector](#)
- [Euclidean Space](#)
- [Linear Algebra](#)
- [Affine](#)
- [Convex](#)
- [Finite Cartesian Product](#)
- [Cartesian Space](#)
- [Determinants](#)
- [Elementary Topology](#)
- [Abstract Topology](#)
- [Abstract Topology 2](#)
- [Connected](#)
- [Abstract Limits](#)
- [Metric Arith](#)
  - [File <metric\\_arith.ML>](#)
- [Elementary Metric Spaces](#)



# Isabelle – A Quick Introduction

Theory dependencies in the Analysis library

[https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/HOL-Analysis/session\\_graph.pdf](https://www.cl.cam.ac.uk/research/hvg/Isabelle/dist/library/HOL/HOL-Analysis/session_graph.pdf)



# Example of a structured proof in Isabelle/HOL

(from Theory Weierstrass\_Theorems in the Isabelle Analysis Library)

```
lemma has_vector_derivative_polynomial_function:
  fixes p :: "real  $\Rightarrow$  'a::euclidean_space"
  assumes "polynomial_function p"
  obtains p' where "polynomial_function p'" " $\wedge x. (p \text{ has\_vector\_derivative } (p' x)) \text{ (at } x\text{)''$ "
proof -
  { fix b :: 'a
    assume "b  $\in$  Basis"
    then
      obtain p' where p': "real_polynomial_function p'" and pd: " $\wedge x. ((\lambda x. p x \bullet b) \text{ has\_real\_derivative } p' x) \text{ (at } x\text{)''$ "
        using assms [unfolded polynomial_function_iff_Basis_inner] has_real_derivative_polynomial_function
        by blast
      have "polynomial_function ( $\lambda x. p' x *_{\mathbb{R}} b$ )"
        using <b  $\in$  Basis> p' const [where 'a=real and c=0]
        by (simp add: polynomial_function_iff_Basis_inner inner_Basis)
      then have " $\exists q. \text{polynomial\_function } q \wedge (\forall x. ((\lambda u. (p u \bullet b) *_{\mathbb{R}} b) \text{ has\_vector\_derivative } q x) \text{ (at } x\text{)''$ "
        by (fastforce intro: derivative_eq_intros pd)
    }
  then obtain qf where qf:
    " $\wedge b. b \in \text{Basis} \implies \text{polynomial\_function } (qf b)$ "
    " $\wedge b x. b \in \text{Basis} \implies ((\lambda u. (p u \bullet b) *_{\mathbb{R}} b) \text{ has\_vector\_derivative } qf b x) \text{ (at } x\text{)''$ "
    by metis
  show ?thesis
proof
  show " $\wedge x. (p \text{ has\_vector\_derivative } (\sum_{b \in \text{Basis}} qf b x)) \text{ (at } x\text{)''$ "
    apply (subst euclidean_representation_sum_fun [of p, symmetric])
    by (auto intro: has_vector_derivative_sum qf)
qed (force intro: qf)
qed
```

[Home](#)[Topics](#)[Download](#)[Help](#)[Submission](#)[Statistics](#)[About](#)

# Archive of Formal Proofs

The Archive of Formal Proofs is a collection of proof libraries, examples, and larger scientific developments, mechanically checked in the theorem prover **Isabelle**. It is organized in the way of a scientific journal, is indexed by **dblp** and has an ISSN: 2150-914x. Submissions are refereed and we encourage companion AFP submissions to conference and journal publications. To cite an entry, please use the **preferred citation style**.

A **development version** of the archive is available as well.

## 2024

- |   |        |
|---|--------|
| <b>Residuated Transition Systems II: Categorical Properties</b><br>by <b>Eugene W. Stark</b>        | Jun 16 |
| <b>Enriched Category Basics</b><br>by <b>Eugene W. Stark</b>  | Jun 16 |
| <b>Alpha-Beta Pruning</b><br>by <b>Tobias Nipkow</b>  | Jun 13 |
| <b>A Preprocessor for Linear Diophantine Equalities and Inequalities</b><br>by <b>René Thiemann</b> | Jun 13 |
| <b>Formalizing Coppersmith's Method</b><br>by <b>Katherine Kosaian</b> and <b>Yong Kiam Tan</b>     | Jun 10 |
| <b>The Riesz Representation Theorem</b><br>by <b>Michikazu Hirata</b>                               | Jun 04 |

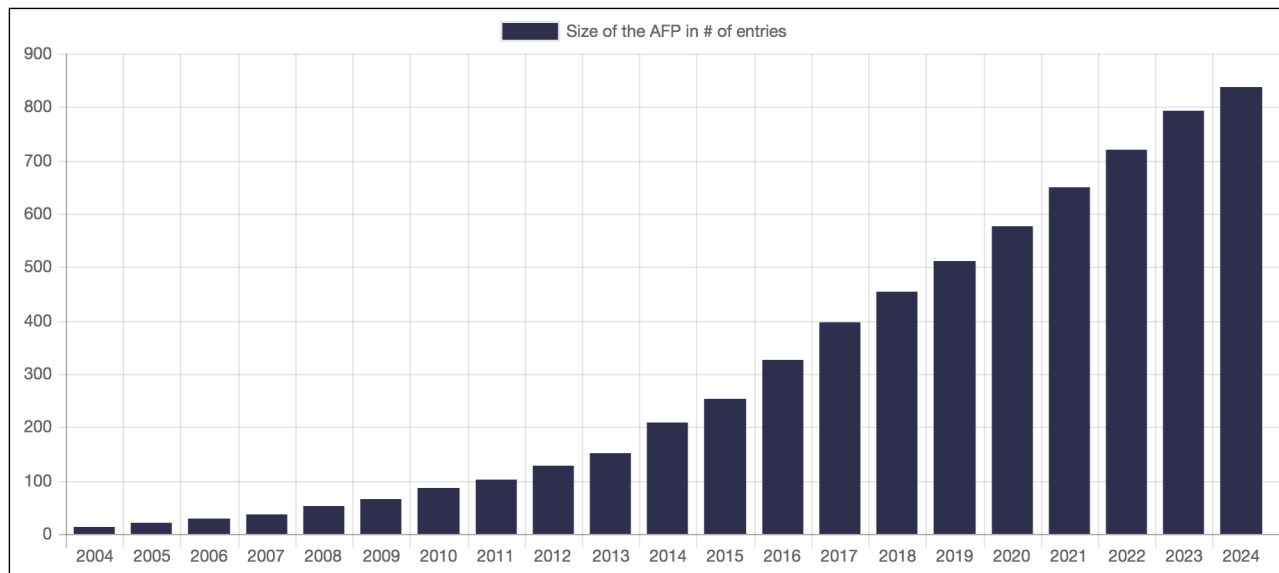
# Isabelle – A Quick Introduction

The Archive of Formal Proofs

A vast collection of formalised material in Mathematics, Computer Science and Logic.



Growth in number of entries:



As of 24 June 2024:

Number of Entries: 837

Number of Authors: 502

Number of Lemmas:

~272,000



Lines of Code:

~4,410,000

# SERAPIS: A concept-oriented search engine for the Isabelle libraries and AFP

By Yiannos Stathopoulos and A. K.-A.

← → ↻ behemoth.cl.cam.ac.uk/search/ 📄 ☆ 📄 👤 ⋮

  Menu ▾

## Welcome to SERAPIS

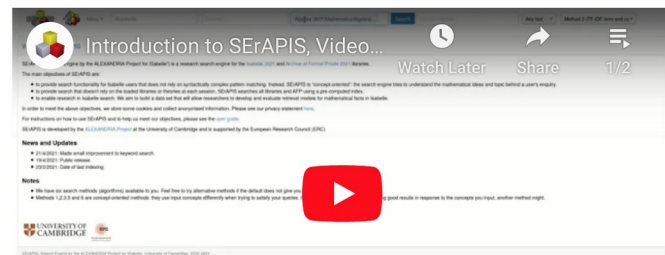
SERAPIS ("Search Engine by the ALEXANDRIA Project for ISabelle") is a research search engine for the [Isabelle 2021](#) and [Archive of Formal Proofs 2021](#) libraries.

The main objectives of SERAPIS are:

- to provide search functionality for Isabelle users that does not rely on syntactically complex pattern matching. Instead, SERAPIS is "concept-oriented": the search engine tries to understand the mathematical ideas and topic behind a user's enquiry.
- to provide search that doesn't rely on the loaded libraries or theories at each session. SERAPIS searches all libraries and AFP using a pre-computed index.
- to enable research in Isabelle search. We aim to build a data set that will allow researchers to develop and evaluate retrieval models for mathematical facts in Isabelle.

In order to meet the above objectives, we store some cookies and collect anonymised information. Please see our [privacy statement](#) [here](#).

We have prepared two short videos to get you started with using SERAPIS:



The main objectives of SERAPIS are:

- to provide search functionality for Isabelle users that does not rely on syntactically complex pattern matching. Instead, SERAPIS is "concept-oriented": the search engine tries to understand the mathematical ideas and topic behind a user's enquiry.
- to provide search that doesn't rely on the loaded libraries or theories at each session. SERAPIS searches all libraries and AFP using a pre-computed index.
- to enable research in Isabelle search. We aim to build a data set that will allow researchers to develop and evaluate retrieval models for mathematical facts in Isabelle.

In order to meet the above objectives, we store some cookies and collect anonymised information. Please see our [privacy statement](#) [here](#).

For attachments on how to use SERAPIS and how to make our objectives, please see the [user guide](#).


SERAPIS is developed by the [ALEXANDRIA Project](#) at the University of Cambridge and is supported by the European Research Council (ERC).

**News and Updates**

- 21/6/2021: Made email representation to keyword search.
- 19/6/2021: Update release.
- 20/2/2021: Date of last release.

**Notes**

- We have six search methods (algorithms) available to you. Feel free to try alternative methods if the default does not give the results you want. The search methods are: Method 1 (Default), Method 2, Method 3, Method 4, Method 5, Method 6, Method 7, Method 8, Method 9, Method 10, Method 11, Method 12, Method 13, Method 14, Method 15, Method 16, Method 17, Method 18, Method 19, Method 20, Method 21, Method 22, Method 23, Method 24, Method 25, Method 26, Method 27, Method 28, Method 29, Method 30, Method 31, Method 32, Method 33, Method 34, Method 35, Method 36, Method 37, Method 38, Method 39, Method 40, Method 41, Method 42, Method 43, Method 44, Method 45, Method 46, Method 47, Method 48, Method 49, Method 50, Method 51, Method 52, Method 53, Method 54, Method 55, Method 56, Method 57, Method 58, Method 59, Method 60, Method 61, Method 62, Method 63, Method 64, Method 65, Method 66, Method 67, Method 68, Method 69, Method 70, Method 71, Method 72, Method 73, Method 74, Method 75, Method 76, Method 77, Method 78, Method 79, Method 80, Method 81, Method 82, Method 83, Method 84, Method 85, Method 86, Method 87, Method 88, Method 89, Method 90, Method 91, Method 92, Method 93, Method 94, Method 95, Method 96, Method 97, Method 98, Method 99, Method 100.




SERAPIS: Search Engine by the ALEXANDRIA Project for Isabelle, University of Cambridge, 2021-2022.

Watch on 




Please visit our YouTube channel for short demo videos, also see our user manual.



## SErAPIS Isabelle Search Engine

7 subscribers

HOME VIDEOS PLAYLISTS CHANNELS ABOUT



### Introducing SErAPIS


(Search Engine by the Alexandria Project for Isabelle)

SErAPIS search engine URLs:  
<https://behemoth.cl.cam.ac.uk/search/>  
[https://behemoth.cl.cam.ac.uk/search/SErAPIS\\_online\\_user\\_guide.pdf](https://behemoth.cl.cam.ac.uk/search/SErAPIS_online_user_guide.pdf)

Yiannos Stathopoulos,  
Angeliki Koutsoukou-Argyraki and  
Lawrence C. Paulson

Department of Computer Science and Technology  
University of Cambridge

Supported by the ERC Advanced Grant ALEXANDRIA, Project 742178  
<https://www.cl.cam.ac.uk/~lp15/Grants/Alexandria/>



European Research Council  
Established by the European Commission


### Welcome to the SErAPIS Isabelle Search Engine channel

47 views • 1 year ago


Introduction to the channel and the SErAPIS Isabelle search engine.

The search engine: <https://behemoth.cl.cam.ac.uk/search/>  
User guide: <https://behemoth.cl.cam.ac.uk/search/...>


Uploads ▶ PLAY ALL



**Introduction to SErAPIS, Video 2: Search Example an...**  
50 views • 1 year ago



**Introduction to SErAPIS, Video 1: Search Controls**  
93 views • 1 year ago



**Welcome to the SErAPIS Isabelle Search Engine...**  
47 views • 1 year ago



## Isabelle Quick Access Links

Quick link: `isabelle.systems/<code>`, e.g. `isabelle.systems/doc`

- [home](#): The official website and download page.

## Communication

- [dev-email](#): The Isabelle development e-mail list.
- [dev](#): Isabelle development hub hosting the repository, ongoing tasks, build status information, etc.
- [email](#): The Isabelle users e-mail list.
- [zulip](#): Real-time discussion platform to exchange ideas, ask questions, and collaborate on Isabelle projects.

## Infrastructure

- [build](#): Build status information including performance statistics and graphs.
- [ci](#): Isabelle/Jenkins continuous integration service.
- [repo](#): The development repository.

# The Isabelle mailing list

# A friendly online community of Isabelle users

lists.cam.ac.uk/sympa/subscribe/cl-isabelle-users



Home

Search for List(s)

Support

Login using Raven

Login locally

## cl-isabelle-users - Isabelle Users List

### List Options

Owners: Lawrence Paulson

Moderators: gerwin.klein, Lawrence Paulson, W. Li

Contact owners

List Home

Subscribe

Unsubscribe

Archive

Post

cl-isabelle-users@lists.cam.ac.uk

Subject: Isabelle Users List

You've made a subscription request to cl-isabelle-users. To confirm your request, please click the button below:

Your e-mail address:

Name:

I subscribe to list cl-isabelle-users

(from early beginners to experts) open to everyone

# Log in to Zulip



Isabelle

<https://isabelle.zulipchat.com>


A cool place for beginners and experts alike playing with mathematics and algorithms in the Isabelle theorem prover!

[View without an account](#)

OR

Email

Password



**Also: Isabelle Zulip chat and**



# Lawrence Paulson's Blog:

← → ↻ [lawrencecpaulson.github.io](https://lawrencecpaulson.github.io)

## Machine Logic

At the junction of computation, logic and mathematics

The formal verification of computer systems has become practical. It has an essential role in tech firms such as Amazon, AMD, Intel, Microsoft and Nvidia. In recent years, researchers have started asking whether verification technology could also benefit research mathematicians. Here, we explore every aspect of doing logic on the computer: its foundations, its applications and the issues involved with formalising mathematics.

### Archive

[general](#) [examples](#) [Isabelle](#) [logic](#) [Isar](#) [Kurt\\_Gödel](#) [set\\_theory](#) [David\\_Hilbert](#)  
[Archive\\_of\\_Formal\\_Proofs](#) [philosophy](#) [newbies](#) [NG\\_de\\_Bruijn](#) [Martin-Löf\\_type\\_theory](#) [verification](#)

# Lawrence Paulson's course material:

← → ↻ cl.cam.ac.uk/teaching/1718/L21/materials.html



Search

Contact us | A-Z | Advanced search

## Department of Computer Science and Technology

Computer Laboratory > Teaching > Courses 2017–18 > Interactive Formal Verification > Course materials

Advanced Operating Systems

Advanced topics in mobile and sensor systems and data modelling

Affective Computing

Algebraic Path Problems

Category Theory, Type Theory and Logic

Chip Multiprocessors

Computer Security: Principles and Foundations

Computer Vision

Interactive Formal Verification

Introduction to Natural Language Syntax and Parsing

Introduction to networking and systems measurements

Large-scale data processing and optimisation

Machine Learning and Algorithms for Data Mining

### Course pages 2017–18

## Interactive Formal Verification

Syllabus

Course materials

Assessment

### Course texts

The primary course text is "Concrete Semantics" by T. Nipkow and G. Klein, and is freely available [here](#). A stripped-down version of this book is supplied with the Isabelle distribution (click the "Documentation" button on the right, and then open "prog-prove" under the "Tutorials" heading), and is called "Programming and proving with Isabelle/HOL". Some material is common to both, but numbered differently. I will reference both.

The additional technical manuals also found under the "Tutorials" heading in the Isabelle documentation panel are also suggested reading for anybody who wishes to advance their usage of Isabelle. The older (but still relevant) Isabelle tutorial, found in the "Documentation" panel under the "Old tutorials" heading also handles some subjects in more detail than "Programming and proving with Isabelle/HOL".

Students are also encouraged to read through existing Isabelle formalisations to learn idioms and tricks from experienced Isabelle users. A good source of vetted Isabelle formalisations, covering a range of subjects in pure mathematics and computer science, is the [Archive of Formal Proofs](#).

### Installing Isabelle

Isabelle should be installed on laboratory machines. However, if you wish to install Isabelle on your own machine, then (assuming a Linux install):

# Aristotle's Assertoric Syllogistic

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on:

<https://plato.stanford.edu/entries/aristotle-logic/>

\* Formal Proof Development: Angeliki Koutsoukou-Argyraiki; *Aristotle's Assertoric Syllogistic*, Archive of Formal Proofs, first published 08/10/2019, available on:

[https://www.isa-afp.org/entries/Aristotles\\_Assertoric\\_Syllogistic.html](https://www.isa-afp.org/entries/Aristotles_Assertoric_Syllogistic.html)

(Only ~200 lines of Isar code!)

Back to the origins :-)

# Aristotle's Assertoric Syllogistic

Syllogisms are structures of sentences each of which can meaningfully be called true or false (assertions “apophanseis”).

A deduction is speech (logos) in which, certain things having been supposed, something different from those supposed results of necessity because of their being so. (Prior Analytics I.2, 24b18–20).



# Aristotle's Assertoric Syllogistic

Assertions (apophanseis): every such sentence must have the same structure:  
Subject (individual/universal); predicate (only universal); must either affirm or deny the predicate of the subject.

Aristotle treats individual predications and general predications as similar in logical form (“Socrates is an animal”, “Humans are animals”).

When the subject is a universal, predication can be either universal or particular.

# Aristotle's Assertoric Syllogistic

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: <https://plato.stanford.edu/entries/aristotle-logic/>

	Affirmations		Denials	
<b>Universal</b>	$P$ affirmed of all of $S$	Every $S$ is $P$ , All $S$ is (are) $P$	$P$ denied of all of $S$	No $S$ is $P$
<b>Particular</b>	$P$ affirmed of some of $S$	Some $S$ is (are) $P$	$P$ denied of some of $S$	Some $S$ is not $P$ , Not every $S$ is $P$
<b>Indefinite</b>	$P$ affirmed of $S$	$S$ is $P$	$P$ denied of $S$	$S$ is not $P$

	Abbreviation	Sentence
Q	$Aab$	$a$ belongs to all $b$ (Every $b$ is $a$ )
	$Eab$	$a$ belongs to no $b$ (No $b$ is $a$ )
	$Iab$	$a$ belongs to some $b$ (Some $b$ is $a$ )
Z	$Oab$	$a$ does not belong to all $b$ (Some $b$ is not $a$ )

```

definition universal_affirmation :: "'a set ⇒ 'a set ⇒ bool" (infixr "Q" 80)
  where "A Q B ≡ ∀ b ∈ B . b ∈ A "

definition universal_denial :: "'a set ⇒ 'a set ⇒ bool" (infixr "E" 80)
  where "A E B ≡ ∀ b ∈ B. ( b ∉ A) "

definition particular_affirmation :: "'a set ⇒ 'a set ⇒ bool" (infixr "I" 80)
  where "A I B ≡ ∃ b ∈ B. ( b ∈ A) "

definition particular_denial :: "'a set ⇒ 'a set ⇒ bool" (infixr "Z" 80)
  where "A Z B ≡ ∃ b ∈ B. ( b ∉ A) "

text< The above four definitions are known as the "square of opposition".>

definition indefinite_affirmation :: "'a set ⇒ 'a set ⇒ bool" (infixr "QI" 80)
  where "A QI B ≡ (( ∀ b ∈ B. (b ∈ A)) ∨ (∃ b ∈ B. (b ∈ A))) "

definition indefinite_denial :: "'a set ⇒ 'a set ⇒ bool" (infixr "EZ" 80)
  where "A EZ B ≡ (( ∀ b ∈ B. (b ∉ A)) ∨ (∃ b ∈ B. (b ∉ A))) "

```

(Note: Aristotle would never consider A to be an 1-element set)

$Eab \rightarrow Eba$

$Iab \rightarrow Iba$

$Aab \rightarrow Iba$

```
lemma aristo_conversion1 :  
  assumes "A E B" shows "B E A"  
  using assms universal_denial_def by blast
```

```
lemma aristo_conversion2 :  
  assumes "A I B" shows "B I A"  
  using assms unfolding particular_affirmation_def  
  by blast
```

```
lemma aristo_conversion3 : assumes "A Q B" and "B ≠{}" shows "B I A"  
  using assms  
  unfolding universal_affirmation_def particular_affirmation_def by blast
```

# Aristotle's Assertoric Syllogistic: the Deductions in the Figures (Moods)

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: <https://plato.stanford.edu/entries/aristotle-logic/>

	First Figure		Second Figure		Third Figure	
	Predicate	Subject	Predicate	Subject	Predicate	Subject
Premise	<i>a</i>	<i>b</i>	<i>a</i>	<i>b</i>	<i>a</i>	<i>c</i>
Premise	<i>b</i>	<i>c</i>	<i>a</i>	<i>c</i>	<i>b</i>	<i>c</i>
Conclusion	<i>a</i>	<i>c</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>

# Aristotle's Assertoric Syllogistic: the Deductions in the Figures (Moods)

Form	Mnemonic	Proof
FIRST FIGURE		
$Aab, Abc \vdash Aac$	<i>Barbara</i>	Perfect
$Eab, Abc \vdash Eac$	<i>Celarent</i>	Perfect
$Aab, Ibc \vdash Iac$	<i>Darii</i>	Perfect; also by impossibility, from <i>Camestres</i>
$Eab, Ibc \vdash Oac$	<i>Ferio</i>	Perfect; also by impossibility, from <i>Cesare</i>
SECOND FIGURE		
$Eab, Aac \vdash Ebc$	<i>Cesare</i>	$(Eab, Aac) \rightarrow (Eba, Aac) \vdash_{Cel} Ebc$
$Aab, Eac \vdash Ebc$	<i>Camestres</i>	$(Aab, Eac) \rightarrow (Aab, Eca) \vdash_{Cel} Ecb \rightarrow Ebc$ $= (Eca, Aab)$
$Eab, Iac \vdash Obc$	<i>Festino</i>	$(Eab, Iac) \rightarrow (Eba, Iac) \vdash_{Fer} Obc$
$Aab, Oac \vdash Obc$	<i>Baroco</i>	$(Aab, Oac + Abc) \vdash_{Imp} Obc$ $\vdash_{Bar} (Aac, Oac)$
THIRD FIGURE		
$Aac, Abc \vdash Iab$	<i>Darapti</i>	$(Aac, Abc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Eac, Abc \vdash Oab$	<i>Felapton</i>	$(Eac, Abc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$
$Iac, Abc \vdash Iab$	<i>Disamis</i>	$(Iac, Abc) \rightarrow (Ica, Abc) \vdash_{Dar} Iba \rightarrow Iab$ $= (Abc, Ica)$
$Aac, Ibc \vdash Iab$	<i>Datisi</i>	$(Aac, Ibc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Oac, Abc \vdash Oab$	<i>Bocardo</i>	$(Oac, +Aab, Abc) \vdash_{Imp} Oab$ $\vdash_{Bar} (Aac, Oac)$
$Eac, Ibc \vdash Oab$	<i>Ferison</i>	$(Eac, Ibc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$

Table of the Deductions in the Figures

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, first published 18/3/2000, substantive revision 17/2/2017, available on: <https://plato.stanford.edu/entries/aristotle-logic/>

# Aristotle's Assertoric Syllogistic: the Deductions in the Figures ("Moods")

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy, <https://plato.stanford.edu/entries/aristotle-logic/>

Form	Mnemonic	Proof
FIRST FIGURE		
$Aab, Abc \vdash Aac$	<i>Barbara</i>	Perfect
$Eab, Abc \vdash Eac$	<i>Celarent</i>	Perfect
$Aab, Ibc \vdash Iac$	<i>Darii</i>	Perfect; also by impossibility, from <i>Camestres</i>
$Eab, Ibc \vdash Oac$	<i>Ferio</i>	Perfect; also by impossibility, from <i>Cesare</i>

subsubsection<First Figure>

**Lemma** Barbara:

```
  assumes "A Q B" and "B Q C" shows "A Q C"
by (meson assms universal_affirmation_def)
```

**Lemma** Celarent:

```
  assumes "A E B" and "B Q C" shows "A E C"
by (meson assms universal_affirmation_def universal_denial_def)
```

**Lemma** Darii:

```
  assumes "A Q B" and "B I C" shows "A I C"
by (meson assms particular_affirmation_def universal_affirmation_def)
```

**Lemma** Ferio:

```
  assumes "A E B" and "B I C" shows "A Z C"
by (meson assms particular_affirmation_def particular_denial_def universal_denial_def)
```

```
text<Example of a deduction with general predication.>
```

```
lemma GreekMortal :
```

```
  assumes "Mortal Q Human" and "Human Q Greek "
```

```
  shows " Mortal Q Greek "
```

```
using assms Barbara by auto
```

```
text<Example of a deduction with individual predication.>
```

```
lemma SocratesMortal:
```

```
  assumes "Socrates ∈ Human " and "Mortal Q Human "
```

```
  shows "Socrates ∈ Mortal "
```

```
using assms by (simp add: universal_affirmation_def)
```



## SECOND FIGURE

$Eab, Aac \vdash Ebc$	<i>Cesare</i>	$(Eab, Aac) \rightarrow (Eba, Aac) \vdash_{Cel} Ebc$
$Aab, Eac \vdash Ebc$	<i>Camestres</i>	$(Aab, Eac) \rightarrow (Aab, Eca) \vdash_{Cel} Ecb \rightarrow Ebc$ $= (Eca, Aab)$
$Eab, Iac \vdash Obc$	<i>Festino</i>	$(Eab, Iac) \rightarrow (Eba, Iac) \vdash_{Fer} Obc$
$Aab, Oac \vdash Obc$	<i>Baroco</i>	$(Aab, Oac + Abc) \vdash_{Imp} Obc$ $\vdash_{Bar} (Aac, Oac)$

\* Source: Robin Smith;  
*Aristotle's Logic*, Stanford  
Encyclopedia of  
Philosophy,  
[https://plato.stanford.edu/  
entries/aristotle-logic/](https://plato.stanford.edu/entries/aristotle-logic/)

subsection <Second Figure>

**lemma** Cesare:

assumes "A E B " and "A Q C" shows "B E C"  
using Celarent aristo\_conversion1 assms by blast

**lemma** Camestres:

assumes "A Q B " and "A E C" shows "B E C "  
using Cesare aristo\_conversion1 assms by blast

**lemma** Festino:

assumes "A E B " and "A I C" shows "B Z C "  
using Ferio aristo\_conversion1 assms by blast

**lemma** Baroco:

assumes "A Q B " and "A Z C" shows "B Z C "  
by (meson assms particular\_denial\_def universal\_affirmation\_def)

THIRD FIGURE

$Aac, Abc \vdash Iab$	<i>Darapti</i>	$(Aac, Abc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Eac, Abc \vdash Oab$	<i>Felapton</i>	$(Eac, Abc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$
$Iac, Abc \vdash Iab$	<i>Disamis</i>	$(Iac, Abc) \rightarrow (Ica, Abc) \vdash_{Dar} Iba \rightarrow Iab$ $= (Abc, Ica)$
$Aac, Ibc \vdash Iab$	<i>Datisi</i>	$(Aac, Ibc) \rightarrow (Aac, Icb) \vdash_{Dar} Iab$
$Oac, Abc \vdash Oab$	<i>Bocardo</i>	$(Oac, +Aab, Abc) \vdash_{Imp} Oab$ $\vdash_{Bar} (Aac, Oac)$
$Eac, Ibc \vdash Oab$	<i>Ferison</i>	$(Eac, Ibc) \rightarrow (Eac, Icb) \vdash_{Fer} Oab$

\* Source: Robin Smith; *Aristotle's Logic*, Stanford Encyclopedia of Philosophy,  
<https://plato.stanford.edu/entries/aristotle-logic/>

### subsubsection <Third Figure>

**Lemma** Darapti:

```
assumes "A Q C " and "B Q C" and "C ≠{}" shows "A I B "  
using Darii assms unfolding universal_affirmation_def particular_affirmation_def  
by blast
```

**Lemma** Felapton:

```
assumes "A E C" and "B Q C" and "C ≠{}" shows "A Z B"  
using Festino aristo_conversion1 aristo_conversion3 assms by blast
```

**Lemma** Disamis:

```
assumes "A I C" and "B Q C" shows "A I B"  
using Darii aristo_conversion2 assms by blast
```

**Lemma** Datisi:

```
assumes "A Q C" and "B I C" shows "A I B"  
using Disamis aristo_conversion2 assms by blast
```

**Lemma** Bocardo:

```
assumes "A Z C" and "B Q C" shows "A Z B"  
by (meson assms particular_denial_def universal_affirmation_def)
```

**Lemma** Ferison:

```
assumes "A E C " and "B I C" shows "A Z B "  
using Ferio aristo_conversion2 assms by blast
```

# Aristotle's Assertoric Syllogistic

A metatheorem by Aristotle:

All deductions can be reduced to Barbara/ Celarent.

# Observations

1) Using Isabelle's automation (Sledgehammer),  
the proofs of the deductions in the Figures are straightforward (one-line)  
The de Bruijn factor would be  $< 1$  !

Example: Compare

**lemma** Camestres:

```
  assumes "A Q B " and "A E C" shows "B E C "  
using Cesare aristo_conversion1 assms by blast
```

(note: Cesare  
reduces to Celarent)

with the original proof:

## Aristotle's proof of Camestres

$$\left| \begin{array}{l} Aab, Eac \vdash Ebc \end{array} \right| \begin{array}{l} \text{Camestres} \\ = (Eca, Aab) \end{array} \left| \begin{array}{l} (Aab, Eac) \rightarrow (Aab, Eca) \vdash_{Cel} Ecb \rightarrow Ebc \end{array} \right.$$

“If a belongs to every b (:= every b is a) but to no c (:=no c is a), then neither will b belong to any c (:=no c is b). For if a belongs to no c (:= no c is a) , then neither does c belong to any a (:= no a is c); but a belonged to every b (:=every b is a); therefore, c will belong to no b (:= no b is c) (for the first figure has come about). And since the privative converts, neither will b belong to any c (:=no c is b).”

Written as:

(1) Aab, (2) Eac, To prove: Ebc.

(3) Eac (from (2))

(4) Eca (from (3) and conversion)

(5) Aab (from (1))

(6) Ecb (from (4), (5) and Celarent)

(7) Ebc (from (6) and conversion)

# Observations

2) The metatheorem that all deductions can be reduced to Barbara/ Celarent can be seen easily from the formal proofs:

**subsection** <Metatheoretical comments>

**text** <The following are presented to demonstrate one of Aristotle's metatheoretical explorations. Namely, Aristotle's metatheorem that: "All deductions in all three Figures can eventually be reduced to either Barbara or Celarent" is demonstrated by the proofs below and by considering the proofs from the previous subsection. >

**lemma** Darii\_reducedto\_Camestres:

```
assumes "A Q B " and "B I C" and "A E C " (*assms, concl. of Darii and A E C *)
shows "A I C"
```

**proof-**

```
have "B E C" using Camestres < A Q B > <A E C> by blast
show ?thesis using < B I C > <B E C>
  by (simp add: particular_affirmation_def universal_denial_def)
```

**qed**

**text** <It is already evident from the proofs in the previous subsection that:

Camestres can be reduced to Cesare.

Cesare can be reduced to Celarent.

Festino can be reduced to Ferio.>

**Lemma** Ferio\_reducedto\_Cesare: **assumes**

"A E B " **and** "B I C" **and** "A Q C " (\*assms, concl. of Ferio and A Q C \*)

**shows** "A Z C"

**proof**-

**have** "B E C" **using** Cesare <A E B > <A Q C> **by** blast

**show** ?thesis **using** <B I C > <B E C>

**by** (simp **add**: particular\_affirmation\_def universal\_denial\_def)

**qed**



**Lemma** Baroco\_reducedto\_Barbara :

**assumes** "A Q B " **and** " A Z C " **and** " B Q C "

**shows** "B Z C" (\*assms , concl. of Baroco and B Q C \*)

**proof-**

**have** "A Q C" **using** <A Q B > < B Q C > Barbara **by** blast

**show** ?thesis **using** <A Q C> < A Z C >

**by** (simp **add:** particular\_denial\_def universal\_affirmation\_def)

**qed**

**Lemma** Bocardo\_reducedto\_Barbara :

**assumes** " A Z C" **and** "B Q C" **and** "A Q B"

**shows** "A Z B" (\*assms, concl of Bocardo and A Q B \*)

**proof-**

**have** "A Q C" **using** <B Q C> < A Q B > **using** Barbara **by** blast

**show** ?thesis **using** <A Q C> < A Z C >

**by** (simp **add:** particular\_denial\_def universal\_affirmation\_def)

**qed**

**text**<Finally, it is already evident from the proofs in the previous subsection that :

Darapti can be reduced to Darii.

Felapton can be reduced to Festino.

Disamis can be reduced to Darii.

Datisi can be reduced to Disamis.

Ferison can be reduced to Ferio. >

**text**<In conclusion, the aforementioned deductions have thus been shown to be reduced to either Barbara or Celarent as follows:

Baroco  $\rightarrow$  Barbara

Bocardo  $\rightarrow$  Barbara

Felapton  $\rightarrow$  Festino  $\rightarrow$  Ferio  $\rightarrow$  Cesare  $\rightarrow$  Celarent

Datisi  $\rightarrow$  Disamis  $\rightarrow$  Darii  $\rightarrow$  Camestres  $\rightarrow$  Cesare

Darapti  $\rightarrow$  Darii

Ferison  $\rightarrow$  Ferio

>

# Observations

3) The assumption that sets at hand must be nonempty is picked up by Isabelle's counterexample tools. (Example)

```
119 lemma Felapton:  
120   assumes "A E C" and "B Q C" (* and "C ≠ {}" *) shows "A Z B"  
121 (* using Festino aristo_conversion1 aristo_conversion3 assms by blast*)
```

Proof state  Auto update  Search:

```
proof (prove)  
goal (1 subgoal):  
1. A Z B
```

Auto Quickcheck found a counterexample:

```
A = {}  
C = {}  
B = {}
```

**lemma** Felapton:

```
assumes "A E C" and "B Q C" and "C ≠ {}" shows "A Z B"  
using Festino aristo_conversion1 aristo_conversion3 assms by blast
```

**Thank you!**